

Claims

What is claimed is:

1 1. Apparatus for implementing enhanced graphical user interface
2 functions in a graphical debugger comprising:
3 a user interface for operatively controlling a graphical user interface;
4 a loadmap display manager coupled to said user interface for
5 implementing a loadmap function;
6 said user interface responsive to said loadmap display manager for
7 displaying a program loadmap;
8 a custom record display manager coupled to said user interface
9 receiving user inputs and implementing a custom record display function;
10 said user interface responsive to said custom record display manager
11 for displaying user selected customized records.

1 2. Apparatus for implementing enhanced graphical user interface
2 functions as recited in claim 1 further includes a debugger server and
3 wherein said loadmap display manager operatively controls said debugger
4 server for implementing said loadmap function.

1 3. Apparatus for implementing enhanced graphical user interface
2 functions as recited in claim 2 wherein said load map function includes
3 debugger means for reading debug data for a program under debug;
4 examining the program debug data, for generating a list of each source and
5 disassembly file contained in the program under debug, and for generating a
6 list of each object or archive program bound to the program under debug at
7 run time to generate said program loadmap for display.

1 4. Apparatus for implementing enhanced graphical user interface
2 functions as recited in claim 3 wherein said load map function further
3 includes debugger means for identifying program load and unload events,
4 and for dynamically updating said program loadmap for display as loadmap
5 information changes responsive to program load and unload events;
6 whereby said program loadmap enables setting a breakpoint within a user
7 selected address range for one instance of a source file, without setting the
8 breakpoint in other instances of the source file.

1 5. Apparatus for implementing enhanced graphical user interface
2 functions as recited in claim 2 wherein said custom record display manager
3 operatively controls said debugger server for implementing said custom
4 record display function; and said custom record display function includes
5 debugger means for identifying a user selected variable, and user selected
6 fields of the variable to be displayed.

1 6. Apparatus for implementing enhanced graphical user interface
2 functions as recited in claim 5 wherein said custom record display function
3 further includes debugger means for identifying a user selected all variables
4 of this type to be customized, for creating a custom record with the user
5 selected fields and for adding said created custom record to a custom type
6 list for displaying user selected customized records for all variables of this
7 type only with the user selected fields.

1 7. Apparatus for implementing enhanced graphical user interface
2 functions as recited in claim 5 wherein said custom record display function
3 further includes debugger means for creating a custom record with the user
4 selected fields and for adding said created custom record to a variables list
5 for displaying the variable only with the user selected fields.

1 8. A debugger computer program product for implementing
2 enhanced graphical user interface functions in a computer system, said
3 debugger computer program product including instructions executed by the
4 computer system to cause the computer system to perform the steps of:
5 reading debug data for a program under debug;
6 examining said program debug data, generating a list of each source
7 and disassembly file contained in the program under debug, and generating
8 a list of each object or archive program bound to the program under debug
9 at run time to generate said program loadmap for display; and
10 displaying said generated program loadmap on a graphical user
11 interface.

1 9. A debugger computer program product for implementing
2 enhanced graphical user interface functions as recited in claim 8 includes
3 the step of identifying program load and unload events, and dynamically
4 updating said program loadmap for display as loadmap information changes
5 responsive to program load and unload events.

1 10. A debugger computer program product for implementing
2 enhanced graphical user interface functions as recited in claim 9 wherein the
3 step of identifying program load and unload events includes the steps of
4 setting breakpoints at program load and unload entry points in the program
5 under debug; and when debugging said program, identifying an unload
6 breakpoint; and removing all information for an archive program being
7 unloaded from said program loadmap to provide a current program loadmap,
8 and displaying said current program loadmap.

1 11. A debugger computer program product for implementing
2 enhanced graphical user interface functions as recited in claim 8 includes
3 the step of identifying a user selected variable and identifying user selected
4 fields of the variable to be displayed, and creating a custom record with the
5 user selected fields.

1 12. A debugger computer program product for implementing
2 enhanced graphical user interface functions as recited in claim 11 includes
3 the step responsive to a user selection of all variables of this type to be
4 customized, adding said created custom record to a custom type list for
5 displaying user selected customized records for all variables of this type only
6 with the user selected fields.

1 13. A debugger computer program product for implementing
2 enhanced graphical user interface functions as recited in claim 11 includes
3 the step adding said created custom record to a variables list for displaying
4 the variable only with the user selected fields.

1 14. A debugger computer program product for implementing
2 enhanced graphical user interface functions in a computer system, said
3 debugger computer program product including instructions executed by the
4 computer system to cause the computer system to perform the steps of:
5 identifying a user selected variable and identifying user selected fields
6 of the variable to be displayed,
7 creating a custom record with the user selected fields; and
8 displaying the variable only with the user selected fields.

1 15. A debugger computer program product for implementing
2 enhanced graphical user interface functions as recited in claim 14 wherein
3 the step of displaying the variable only with the user selected fields includes
4 the step adding said created custom record to a variables list for displaying
5 the variable only with the user selected fields.

1 16. A debugger computer program product for implementing
2 enhanced graphical user interface functions as recited in claim 14 includes
3 the step of identifying a user selection of all variables of this type to be
4 customized, adding said created custom record to a custom type list for
5 displaying user selected customized records for all variables of this type only
6 with the user selected fields.

1 17. A method for implementing enhanced graphical user interface
2 functions in a graphical debugger comprising the steps of:
3 utilizing a loadmap display manager for implementing a loadmap
4 function and generating a program loadmap for a program under debug;
5 displaying said generated program loadmap responsive to said
6 loadmap function;
7 utilizing a custom record display manager for receiving user inputs
8 and implementing a custom record display function;
9 displaying user selected customized records responsive to said
10 custom record display function.

1 18. A method for implementing enhanced graphical user interface
2 functions in a graphical debugger as recited in claim 17 wherein the step of
3 utilizing a loadmap display manager for implementing a loadmap function
4 and generating a program loadmap for a program under debug includes the
5 steps of reading debug data for a program under debug; examining said
6 program debug data, generating a list of each source and disassembly file
7 contained in the program under debug, and generating a list of each object
8 or archive program bound to the program under debug at run time to
9 generate said program loadmap for display; and displaying said generated
10 program loadmap on a debugger graphical user interface.

1 19. A method for implementing enhanced graphical user interface
2 functions in a graphical debugger as recited in claim 18 includes the steps of
3 identifying program load and unload events, dynamically updating said
4 program loadmap and displaying said updated loadmap responsive to
5 identified program load and unload events.

1 20. A method for implementing enhanced graphical user interface
2 functions in a graphical debugger as recited in claim 17 wherein the step of
3 utilizing a custom record display manager for receiving user inputs and
4 implementing a custom record display function includes the steps of
5 identifying a user selected variable and identifying user selected fields of the
6 variable to be displayed, creating a custom record with the user selected
7 fields; and displaying the variable only with the user selected fields.

1 21. A method for implementing enhanced graphical user interface
2 functions in a graphical debugger as recited in claim 20 includes the steps of
3 identifying a user selection of all variables of this type to be customized,
4 adding said created custom record to a custom type list for displaying user
5 selected customized records for all variables of this type only with the user
6 selected fields.